Amendments to the Claims:

The following claims will replace all prior versions of the claims in this application (in the unlikely event that no claims follow herein, the previously pending claims will remain):

1. (Currently Amended) An incremental remote loading ~~apparatus~~ system, comprising:

a dependent reader ~~module~~ means for receiving a cross-compiled object file from a program development tool, analyzing the cross-compiled object file according to a type of an object file and detecting independent linking information from the type of the cross-compiled object file; and

an independent linker ~~module~~ means for receiving the detected linking information from said dependent reader ~~module~~ means, downloading the object file to a target system by using the detected linking information and rearranging target ~~modules~~ means of the target system.

2. (Currently Amended) The incremental remote loading ~~apparatus~~ system as recited in claim 1, wherein said dependent reader ~~module~~ means includes:

a COFF reader ~~module~~ means for receiving a cross-compiled COFF object file from said program development tool, analyzing the object file dependently to a COFF type of the object file and detecting independent linking information from the type of the cross-compiled COFF object file; and

an ELF reader ~~module~~ means for receiving a cross-compiled ELF object file from the program development tool, analyzing the object file dependently to an ELF type of the object file and detecting independent linking information from the type of the cross-compiled ELF object file.

3. (Currently Amended) The incremental remote loading ~~apparatus~~ system as recited in claim 1, wherein the linking information includes section information, symbol information and rearrangement information.

4. (Currently Amended) The incremental remote loading ~~apparatus~~ system as recited in claims 3, wherein said linker ~~module~~ means rearranges not only the cross-compiled object files being loaded but also the loaded target ~~modules~~ means and provides

an incremental remote linking, which links object files to the target system without a linking order of the cross-compiled object files.

5. (Currently Amended) An incremental remote loading method, comprising ~~the steps of~~:

a) at a reader module, analyzing necessary linking information for liking object files;

b) at a linker, allocating a target memory space for sections according to a section information;

c) determining whether each entry of a symbol table is defined or not and calculating addresses of sections in a target memory;

d) determining, according to a result of said step c), whether a symbol is defined or not in case said symbol is stored in said symbol table or inserting a new symbol to the symbol table in case the symbol is not in said symbol table and determining whether the new symbol is defined or not;

e) rearranging ~~an~~ the object file if ~~a~~ the symbol is defined or rearranging the object file after transforming a defined symbol in case the symbol is not defined; and

f) transmitting ~~a~~ the rearranged object file to ~~a~~ the target memory.

6. (Currently Amended) The incremental remote loading method as recited in claim 5, wherein said step a) includes ~~the steps of~~:

a1) determining a type of ~~the~~ cross-compiled object file;

a2) analyzing the linking information by connecting a COFF reader in case said object file type is a COFF type; and

a3) analyzing the linking information by connecting an ELF reader in case said object file type is an ELF type.

7. (Currently Amended) The incremental remote loading method as recited in claim 5, wherein said step d) includes ~~the steps of~~:

d1) generating a new symbol if an entry is not stored in said symbol table and is not a defined symbol;

d2) adding symbol information including a symbol name and an address of a target memory to the newly generated symbol; and

d3) registering and inserting the new symbol, which is generated and the symbol information added, to the symbol table.

8. (Currently Amended) The incremental remote loading method as recited claim 5, wherein transforming the undefined symbol to the defined symbol process in said step e) includes ~~the steps of~~:

e1) transforming ~~an~~ the undefined symbol to ~~an~~ the defined symbol if ~~a~~ the symbol in ~~a~~ the symbol table is undefined; and

e2) adding an address of ~~a~~ the target memory and the rearranging modules of a target system by using a rearrangement information of said undefined symbol.

9. (Currently Amended) The incremental remote loading method as recited in claim 5, wherein said step d) includes ~~the steps of~~:

d1) bringing ~~a~~ the rearranged symbol to said symbol table;

d2) rearranging a text and data sections in a host system base on ~~a~~ the rearrangement information of an entry and an address of ~~a~~ the target memory of ~~a~~ the symbol in case said brought symbol is defined; and

d3) adding ~~a~~ the rearrangement information to said symbol incase said brought symbol is undefined.

10. (Currently Amended) A computer-readable recording medium storing instruction for executing an incremental remote loading method, comprising the functions of:

a) analyzing necessary linking information for linking object files by a reader modules;

b) allocating a target memory space for sections according to section information by a linker;

c) determining whether each entry of a symbol table is defined or not and calculating addresses of sections in a target memory;

d) determining, according to a result of said step c), whether a symbol is defined or not in case said symbol is stored in said symbol table or inserting a new symbol to the symbol table in case the symbol is not in said symbol table and determining whether the new symbol is defined or not;

e) rearranging ~~an~~ the object file if ~~a~~ the symbol is defined or rearranging the object file after transforming a defined symbol in ease a symbol is not defined; and

f) transmitting ~~a~~ the rearranged object file to ~~a~~ the target memory.

11. (New) The computer-readable recording medium as recited in claim 10, wherein said step a) includes:

a1) determining a type of cross-compiled object file;

a2) analyzing the linking information by connecting a COFF reader in case said object file type is a COFF type; and

a3) analyzing the linking information by connecting an ELF reader in case said object file type is an ELF type.

12. (New) The computer-readable recording medium as recited in claim 10, wherein said step d) includes:

d1) generating a new symbol if an entry is not stored in said symbol table and is not a defined symbol;

d2) adding symbol information including a symbol name and an address of a target memory to the newly generated symbol; and

d3) registering and inserting the new symbol, which is generated and the symbol information added, to the symbol table.

13. (New) The computer-readable recording medium as recited claim 10, wherein transforming the undefined symbol to the defined symbol process in said step e) includes:

e1) transforming the undefined symbol to the defined symbol if the symbol in the symbol table is undefined; and

e2) adding an address of the target memory and the rearranging modules of a target system by using a rearrangement information of said undefined symbol.

14. (New) The computer-readable recording medium as recited in claim 10, wherein said step d) includes:

d1) bringing the rearranged symbol to said symbol table;

d2) rearranging a text and data sections in a host system base on the rearrangement information of an entry and an address of the target memory of the symbol in case said brought symbol is defined; and

d3) adding the rearrangement information to said symbol incase said brought symbol is undefined.